ABSTRACT

A stochastic gradient implementation of a generalised multi-channel noise reduction scheme, called the Spatially Pre-processed Speech Distortion Weighted Multi-channel Wiener Filter (SP-SDW-MWF), has recently been proposed in [1]. In order to compute a regularisation term in the filter update formulas, data buffers are required in this implementation, resulting in a large memory usage. This paper shows that by approximating this regularisation term in the frequency-domain the memory usage (and the computational complexity) can be reduced drastically. Experimental results demonstrate that this approximation only gives rise to a limited performance difference and that hence the proposed algorithm preserves the robustness benefit of the SP-SDW-MWF over the GSC (with Quadratic Inequality Constraint).

1. INTRODUCTION

Noise reduction algorithms in hearing aids and cochlear implants are crucial for hearing impaired persons to improve speech intelligibility in background noise. Multi-microphone systems exploit spatial in addition to temporal and spectral information of the desired and noise signals and are hence preferred to single-microphone systems. For small-sized arrays such as in hearing instruments, multi-microphone noise reduction however goes together with an increased sensitivity to errors in the assumed model such as microphone mismatch, reverberation, etc.

In [2] a generalised noise reduction scheme, called the Spatially Pre-processed Speech Distortion Weighted Multi-channel Wiener Filter (SP-SDW-MWF), has been proposed (cf. Section 2). It encompasses both the Generalised Sidelobe Canceller (GSC) and the MWF [3, 4] as extreme cases and allows for in-between solutions such as the Speech Distortion Regularised GSC (SDR-GSC). By taking speech distortion explicitly into account in the design criterion of the adaptive stage, the SP-SDW-MWF (and the SDR-GSC) resembles the GSC [5, 6, 7], where the standard adaptive filter has been replaced by an adaptive SDW-MWF.

The desired speaker is assumed to be in front of the microphone array (having M microphones), and an endfire array is used. The fixed beamformer creates a so-called speech reference $y_0[k] = x_0[k] + v_0[k]$ (with $x_0[k]$ and $v_0[k]$ respectively the speech and the noise component of $y_0[k]$) by steering a beam towards the front, whereas the blocking matrix creates $M - 1$ so-called noise references $y_i[k] = x_i[k] + v_i[k], i = 1 \ldots M - 1$, by steering zeroes towards the front. During speech-periods these references consist of speech-noise, i.e. $y_i[k] = x_i[k] + v_i[k]$, whereas during noise-only periods the noise components $v_i[k]$ are observed. We assume that the second-order statistics of the noise are sufficiently stationary such that they can be estimated during noise-only-periods and used during subsequent speech-periods. This requires the use of a voice activity detection (VAD) mechanism.

Let $N$ be the number of input channels to the multi-channel Wiener filter in Figure 1 ($N = M$ if $w_0$ is present, $N = M - 1$ otherwise). Let the FIR filters $w[k]$ have length $L$, and consider the $L$-dimensional data vectors $y[k], x[k], v[k], w[k]$. The NL-dimensional stacked filter $w[k]$ and the NL-dimensional stacked data vector $y[k]$, defined as

$$y[k] = [y_1[k], y_2[k], \ldots, y_M[k] - 1]_T,$$

$$w[k] = [w_1^T[k], w_2^T[k], \ldots, w_M^T[k] - 1]_T,$$

$$y[k] = [y_1^T[k], y_2^T[k], \ldots, y_M^T[k] - 1]_T.$$

Figure 1: Spatially Pre-processed SDW-MWF
adapt the filter $\mathbf{w}[k]$ using (5) during noise-only-periods, based on approximating the regularisation term in (6) by

$$r[k] = \frac{1}{\mu} \left[ \mathbf{y}_b[k] \mathbf{y}_b^T[k] \mathbf{v}[k] - \mathbf{v}[k]^T \mathbf{w}[k] \right],$$

with $\mathbf{y}_b[k]$ a vector from the circular speech+noise-buffer $\mathbf{B}_v$. However, this estimate of $r[k]$ is quite bad, resulting in a large excess error, especially for small $\mu$ and large $\rho'$. Hence, it has been suggested to use an estimate of the average clean speech correlation matrix $\delta' \mathbf{x}[k]^T \mathbf{x}[k]$ in (6), such that $r[k]$ can be computed as

$$r[k] = \frac{1}{\mu} \left( 1 - \lambda \right) \sum_{i=0}^{\infty} \left( \lambda^{k-i} \mathbf{y}_b[l] \mathbf{y}_b^T[l] - \mathbf{v}[l] \mathbf{v}^T[l] \right) \cdot \mathbf{w}[k].$$

with $\lambda$ an exponential weighting factor and the step size $\rho$ in (7) now equal to

$$\rho = \frac{\mathbf{v}^T[k] \mathbf{v}[k] + \frac{1}{\mu} \left( 1 - \lambda \right) \sum_{i=0}^{\infty} \lambda^{k-i} \mathbf{y}_b^T[l] \mathbf{y}_b[l] - \mathbf{v}[l] \mathbf{v}^T[l] + \delta'.$$

For stationary noise a small $\lambda$, i.e. $1/(1-\lambda) \gg NL$, suffices. However, in practice the speech and the noise signals are often spectrally highly non-stationary (e.g., multi-talker babble noise), whereas their long-term spectral and spatial characteristics usually vary more slowly in time. Spectrally highly non-stationary noise can still be spatially suppressed by using an estimate of the long-term correlation matrix in $\mathbf{r}[k]$, i.e. $1/(1-\lambda) \gg NL$.

In order to avoid expensive matrix operations for computing (9), it is assumed in [1] that $\mathbf{w}[k]$ varies slowly in time, i.e. $\mathbf{w}[k] \approx \mathbf{w}[l]$, such that (9) can be approximated without matrix operations as

$$r[k] = \lambda \mathbf{r}[k-1] + \left( 1 - \lambda \right) \frac{1}{\mu} \left[ \mathbf{y}_b[k] \mathbf{y}_b^T[k] - \mathbf{v}[k] \mathbf{v}^T[k] \right] \mathbf{w}[k].$$

However, as will be shown in the next paragraph, this assumption is actually not required in a frequency-domain implementation.

### 3.2 Efficient Frequency-Domain (FD) implementation

In [1] the SG-FD algorithm has been converted to a frequency-domain implementation by using a block-formulation and overlap-save procedures (similar to standard FD adaptive filtering techniques [9]). However, the SG-FD algorithm in [1] (Algorithm 1) requires the storage of large data buffers (with typical buffer lengths $L_y = 10000 \ldots 20000$). A substantial memory (and computational complexity) reduction can be achieved by the following two steps:

• When using (9) instead of (10) for calculating the regularisation term, correlation matrices instead of data buffers need to be stored. The FD implementation of the total algorithm is then summarised in Algorithm 2, where $2L \times 2L$-dimensional speech and noise correlation matrices $\mathbf{S}_i[l]$ and $\mathbf{S}_j[l]$, $i = M-N \ldots M-1$ and $j = M-N \ldots M-1$ are used for calculating the regularisation term $\mathbf{R}_i[k]$ and (part of) the step size $\lambda[k]$. These correlation matrices are updated respectively during speech-periods and noise-only-periods. However, this first step does not necessarily reduce the memory usage ($NL_y$ for data buffers vs. $2NL_y$ for correlation matrices) and will even increase the computational complexity, since the correlation matrices are not diagonal.

• The correlation matrices in the frequency-domain can be approximated by diagonal matrices, since $\mathbf{F} \mathbf{k} \mathbf{F}^H$ in Algorithm 2 can be well approximated by $\mathbf{I}_L/2$ [10]. Hence, the speech and the noise correlation matrices are updated as

$$\mathbf{S}_i[l] = \lambda \mathbf{S}_i[l-1] + (1 - \lambda) \mathbf{V}_i[l] \mathbf{V}_i[l]^H/2,$$

$$\mathbf{S}_i[l] = \lambda \mathbf{S}_i[l-1] + (1 - \lambda) \mathbf{W}_i[l] \mathbf{W}_i[l]^H/2,$$

1 In [1] it has been shown that storing noisy-only-vectors $\mathbf{v}_i[l]$, $i = M-N \ldots M-1$ during noise-only-periods in a circular noise-buffer $\mathbf{B}_v \in \mathbb{R}^{NL_y \times L_y}$ additionally allows adaptation during speech+noise-periods.

2 When using correlation matrices, filter adaptation can only take place during noise-only-periods, since during speech-periods the desired signal $d[k]$ cannot be constructed from the noise-buffer $\mathbf{B}$, any more.
Algorithm 2 FD implementation (without approximation)

initialisation and matrix definitions:

\[
\begin{align*}
W_i = & \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}^T, i = M-N, \ldots, M-1 \\
P_{\text{w}}[0] = & \mathbf{0}, \quad m = 0, \ldots, 2L-1 \\
F = & 2L \times 2L \text{-dimensional DFT matrix} \\
g = & \begin{bmatrix} I_L & 0_L \ 0_L & I_L \end{bmatrix}, \\
\Omega_L = & \text{a } L \times L \text{ matrix with zeros, } I_L = L \times L \text{ identity matrix}
\end{align*}
\]

For each new block of \( L \) samples (per channel):

\[
\begin{align*}
d[k] = & \begin{bmatrix} y_0[kL-\Delta] & \cdots & y_0[kL-\Delta+L-1] \end{bmatrix}^T \\
\end{align*}
\]

Output signal:

\[
\begin{align*}
e[k] = & \mathbf{d}[k] - kF^{-1} \sum_{j=0}^{M-1} \mathbf{Y}_j[k] \mathbf{W}_j[k], \quad \mathbf{E}[k] = Fk^T e[k]
\end{align*}
\]

If speech detected:

\[
\begin{align*}
S_l^f[k] = & \left( 1 - \lambda \right) \sum_{i=0}^{k} \lambda^{k-i} \mathbf{Y}_i^H[l] Fk^T Fk^{-1} \mathbf{Y}_i[l] \\
\end{align*}
\]

If noise detected:

\[
\begin{align*}
S_l^f[k] = & \left( 1 - \lambda \right) \sum_{i=0}^{k} \lambda^{k-i} \mathbf{V}_i^H[l] Fk^T Fk^{-1} \mathbf{V}_i[l] \\
\end{align*}
\]

Update formula (only during noise-only-periods):

\[
\begin{align*}
\mathbf{R}_i[k] = & \frac{1}{\mu} \sum_{j=0}^{M-1} \left[ S_l^f[j] - \mathbf{S}_l^f[j] \right] \mathbf{W}_j[k] \\
\mathbf{W}_i[k+1] = & \mathbf{W}_i[k] + P_{\text{w}} Fg^F \mathbf{A}[k] \left( \mathbf{V}_i^H[k] \mathbf{E}[k] - \mathbf{R}_i[k] \right)
\end{align*}
\]

with

\[
\begin{align*}
\mathbf{A}[k] = & \frac{2\rho^I}{L} \text{diag} \left\{ P_{0}^{-1}[k], \ldots, P_{2L-1}^{-1}[k] \right\} \\
P_{\text{w}} = & \sum_{j=0}^{k} \left( 1 - \gamma \right) \left( P_{\text{w}}[k] + P_{\text{w}}[k] \right) \\
P_{\text{w}}[k] = & \sum_{j=0}^{k} \left( \mathbf{V}_j[k] \mathbf{V}_j[k]^H \right) \mu \sum_{j=0}^{M-1} S_l^f[j] \mathbf{S}_l^f[j]
\end{align*}
\]

leading to a significant reduction in memory usage (and computational complexity), cf. Section 4, while having a minimal impact on the performance and the robustness, cf. Section 5. We will refer to this algorithm as Algorithm 3. This algorithm is in fact quite similar to [11], which is derived directly from a frequency-domain cost function. Some major differences however exist, e.g. in [11] the regularisation term \( R_i[k] \) is absent, the term \( Fg^F \) is approximated by \( I_{2L} / 2 \) and the speech and the noise correlation matrices are block-diagonal.

4. MEMORY AND COMPUTATIONAL COMPLEXITY

Table 1 summarises the computational complexity and the memory usage for the FD implementation of the QIC-GSC (computed using the NLMS-based Scalled Projection Algorithm (SPA) [5]) and the SDW-MWF (Algorithm 1 and 3). The computational complexity is expressed as the number of operations (i.e. real multiplications and additions (MAC) per second) in MIPS and the memory usage is expressed in kWords. We assume that one complex multiplication is equivalent to 4 real multiplications and 2 real additions and that a 2L-point FFT of a real input vector requires 2Llog_2 2L real MACs

```
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Complexity</th>
<th>MIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSC-SPA</td>
<td>(3M - 1)FT + 14M - 12</td>
<td>2.02</td>
</tr>
<tr>
<td>MWF (Algo1)</td>
<td>(3N + 5)FT + 28N + 6</td>
<td>3.10(a) , 4.13(b)</td>
</tr>
<tr>
<td>MWF (Algo3)</td>
<td>(3N + 2)FT + 8N^2 + 14N + 3</td>
<td>2.54(a) , 3.98(b)</td>
</tr>
<tr>
<td>Memory</td>
<td>kWords</td>
<td></td>
</tr>
<tr>
<td>GSC-SPA</td>
<td>4M(2 - L) + 6L</td>
<td>0.45</td>
</tr>
<tr>
<td>MWF (Algo1)</td>
<td>2NL_0 + 6NL + 7L</td>
<td>40.61(a) , 60.80(b)</td>
</tr>
<tr>
<td>MWF (Algo3)</td>
<td>4NL_0^2 + 6NL + 7L</td>
<td>1.12(a) , 1.95(b)</td>
</tr>
</tbody>
</table>
```

(assuming the radix-2 FFT algorithm). From this table we can draw the following conclusions:

- The computational complexity of the SDW-MWF (Algorithm 1) filter with \( w_0 \) is about twice the complexity of the GSC-SPA (and even less without \( w_0 \)). The approximation in the SDW-MWF (Algorithm 3) further reduces the complexity. However, this only remains true for a small number of input channels, since the approximation introduces a quadratic term \( O(N^2) \).
- Due to the storage of the speech-noise-buffer, the memory usage of the SDW-MWF (Algorithm 1) is quite high in comparison with the GSC-SPA (depending on the size of the data buffer \( L_o \) of course). By using the approximation in the SDW-MWF (Algorithm 3), the memory usage can be drastically reduced. Note however that also for the memory usage a quadratic term \( O(N^2) \) is introduced.

5. EXPERIMENTAL RESULTS

In this paragraph it is shown that practically no performance difference exists between implementing the SDW-MWF using Algorithm 1 or 3, such that the SDW-MWF using the proposed implementation preserves its robustness benefit over the GSC (and the QIC-GSC).

5.1 Set-up and performance measures

A 3-microphone BTE has been mounted on a dummy head in an office room. The desired source is positioned in front of the head (at 0°) and consists of English sentences. The noise scenario consists of three multi-talker babble noise sources, positioned at 75°, 180° and 240°. The desired signal and the total noise signal both have a level of 70 dB SPL at the centre of the head. For evaluation purposes, the speech and the noise signal have been recorded separately. In the experiments, the microphones have been calibrated in an anechoic room with the BTE mounted on the head. A delay-and-sum beamformer is used as fixed beamformer \( \mathbf{A}(z) \). The blocking matrix \( B(z) \) pairwise subtracts the time-aligned calibrated microphone signals. The filter length \( L = 32 \), the step size \( \gamma = 0.8 \) and \( L_0 = 0.999 \).

To assess the performance, the intelligibility weighted signal-to-noise ratio improvement \( \Delta \text{SNR}_{\text{intellig}} \) is used, defined as

\[
\Delta \text{SNR}_{\text{intellig}} = \sum_i \left( \text{SNR}_{\text{out},i} - \text{SNR}_{\text{in},i} \right),
\]

where \( i \) expresses the importance for intelligibility of the \( i \)-th one-third octave band with centre frequency \( f_i^o \) [12], and where \( \text{SNR}_{\text{out},i} \) and \( \text{SNR}_{\text{in},i} \) are respectively the output and the input SNR (in dB) in this band. Similarly, we define an intelligibility weighted spectral distortion measure, called \( \Delta \text{SD}_{\text{intellig}} \), of the desired signal as

\[
\Delta \text{SD}_{\text{intellig}} = \sum_i L_i \text{SD}_i,
\]

with \( \text{SD}_i \) the average spectral distortion (dB) in the \( i \)-th one-third band, calculated as

\[
\text{SD}_i = \frac{1}{2^{1/6} - 2^{-1/6}} \int_{f_{i-1/6}}^{f_{i+1/6}} 20 \log_{10} G_z(f) \, df.
\]
with $G_s(f)$ the power transfer function of speech from the input to the output of the noise reduction algorithm. To exclude the effect of the spatial pre-processor, the performance measures are calculated w.r.t. the output of the fixed beamformer, i.e. the speech reference.

5.2 Experimental results

Figures 2 and 3 depict the SNR improvement and the speech distortion of the SP-SDW-MWF (with $w$) and the SDR-GSC (without $w_0$) as a function of the trade-off parameter $\nu$, for Algorithm 1 (no approx) and Algorithm 3 (approx). These figures also depict the effect of a gain mismatch $\nu = 4$ dB at the second microphone. From these figures it can be observed that approximating the regularisation term results in a small performance difference (smaller than 0.5 dB). For some scenarios the performance is even better for Algorithm 3 than for Algorithm 1, probably since in Algorithm 1 it is assumed that the filter $w(f)$ varies slowly in time.

Hence, also when implementing the SDW-MWF using Algorithm 3, it still preserves its robustness benefit over the GSC (and the QIC-GSC). E.g. it can be observed that the GSC (i.e. SDR-GSC with $1/\mu = 0$) will result in a large speech distortion (and a smaller SNR improvement) when microphone mismatch occurs. Both the SDR-GSC and the SDW-MWF add robustness to the GSC, i.e. distortion increases for increasing $1/\mu$. The performance of the SDW-MWF is even hardly effected by microphone mismatch.

6. CONCLUSION

In this paper we have shown that the memory usage (and the computational complexity) of the SDW-MWF can be reduced drastically by approximating the regularisation term in the frequency-domain, i.e. by computing the regularisation term using (diagonal) FD correlation matrices instead of TD data buffers. It has been shown that approximating the regularisation term only results in a small performance difference, such that the robustness benefit of the SDW-MWF is preserved at a smaller computational cost, which is comparable to the NLMS-based implementation for QIC-GSC.

REFERENCES


